

WHAT IS CLAIMED IS:

- 1 1. A computer system for optimizing processing of an annotation
2 request from a client, comprising:
3 a request processor for receiving said annotation request from said client;
4 a task queue for storing a plurality of constituent tasks that need to be
5 performed for said annotation request;
6 a thread-controlling means for maintaining a plurality of threads; and
7 an assigning means for assigning said plurality of threads to said plurality
8 of constituent tasks in said task queue.
- 1 2. A computer system according to claim 1, wherein said plurality of
2 threads is independent from said plurality of constituent tasks stored in said task queue.
- 1 3. A computer system according to claim 1, wherein said plurality of
2 threads is persistent.
- 1 4. A computer system according to claim 1, wherein said plurality of
2 constituent tasks is arranged in a substantially first-in-first-out basis within said task
3 queue.
- 1 5. A computer system according to claim 1, wherein when a thread is
2 available for assignment, said thread is assigned to a constituent task when said
3 constituent task is ready for execution.
- 1 6. A computer system according to claim 5, wherein said assigned
2 thread is released upon conclusion of said constituent task.
- 1 7. A computer system according to claim 1, wherein said plurality of
2 constituent tasks includes checking a cache to determine whether information pertaining
3 to said annotation request is present in said cache.
- 1 8. A computer system according to claim 1, wherein said plurality of
2 constituent tasks includes retrieving information pertaining to said annotation request
3 from one or more sources.
- 1 9. A computer system according to claim 8, wherein said one or more
2 sources include the Internet.

1 10. A computer system according to claim 1, wherein said plurality of
2 constituent tasks includes annotating a retrieved web page with additional hyperlinks.

1 11. A computer system according to claim 1, wherein said plurality of
2 constituent tasks includes updating a cache with annotated information.

1 12. A computer system according to claim 1, further comprising:
2 a I/O queue for storing a plurality of I/O tasks identified from said
3 plurality of constituent tasks, wherein said plurality of I/O tasks only perform input and/or
4 output functions.

1 13. A computer system according to claim 12, wherein two or more of
2 said plurality of I/O tasks are executed in a parallel manner.

1 14. A computer system according to claim 12, wherein said task queue
2 is notified upon completion of each of said plurality of I/O tasks.

1 15. A computer system according to claim 14, wherein upon said
2 notification one or more of said plurality of constituent tasks which require results from
3 said executed I/O tasks are rendered ready for execution.

1 16. A computer system for optimizing processing of an annotation
2 request, comprising:
3 a task queue for storing a plurality of requisite tasks needed to execute said
4 annotation request; and
5 a thread-controlling means for controlling a thread pool having a plurality
6 of threads;
7 wherein said thread-controlling means assigns an available thread from
8 said thread pool to an execution-ready requisite task.

1 17. A computer system according to claim 16, said thread pool is
2 independent of said plurality of requisite tasks.

1 18. A computer system according to claim 16, wherein said assigned
2 thread is released back into said thread pool for subsequent assignment when the
3 execution of said execution-ready requisite task is completed.

1 19. A method for optimizing processing of an annotation request
2 received from a client, comprising the steps of:
3 identifying a plurality of constituent tasks needed to complete the
4 execution of said annotation request;
5 storing said plurality of constituent tasks into a task queue;
6 maintaining a plurality of threads assignable to said plurality of constituent
7 tasks; and
8 assigning an available thread to a constituent task when said constituent
9 task is ready for execution.

1 20. A method according to claim 19, further comprising the steps of:
2 identifying a plurality of I/O tasks from said plurality of constituent tasks;
3 storing said plurality of I/O tasks into an I/O queue; and
4 executing two or more of said plurality of I/O tasks in a parallel manner.

1 21. A method according to claim 20, further comprising the step of:
2 rendering one or more constituent tasks which require results from said
3 executed I/O tasks ready for execution.

1 22. A method according to claim 19, wherein said plurality of threads
2 is persistent.

1 23. A method according to claim 19, wherein said assigning of said
2 available thread to said constituent task is independent of the nature of said constituent
3 task.